```
module memory (address,data_in,data_out,read,write,clk);

input [11:0]address;
input [15:0]data_in;
input clk,read,write;

output[15:0]data_out;

wire [11:0]address;
wire [15:0]data_in;
wire clk,read,write;



reg [15:0]data_out;
reg [15:0] Memory [0:4095];

always @(posedge clk)
begin

//writting data
if(write)
begin
Memory[address] <= data_in;
end

// reading data
if(read)
begin
data_out= Memory[address];
end

end
endmodule


module memory_tb (address,data_in,data_out,read,write,clk);

output [11:0]address;
output [15:0]data_in;
output clk,read,write;

input[15:0]data_out;

reg clk,read,write;
reg [15:0]data_in;
reg [11:0]address;

wire [15:0]data_out;
```

```
initial
begin
$dumpfile("memory_tb.vcd");
$dumpvars;
address= 12'b000000001111;
data_in= 16'b0000000011110000;
read= 0;
write= 0;
clk= 0;
#250 $finish;
end

// writting data values
initial
begin
#40
write = 1;
read = 0;
data_in = 16'b0000000000001111;
address= 12'b000001101100;
#20 data_in= 16'b0000000011010000;
address= 12'b000000111000;
end

// reading data values
initial
begin
#80
read = 1;
write = 0;
address= 12'b000000111000;
#20 address= 12'b000001101100;
end


always
begin
#5 clk =~clk;
end

memory M1 (address,data_in,data_out,read,write,clk);

endmodule
```